



Revista UNILUS Ensino e Pesquisa
v. 21, n. 63, abr./jun. 2024
ISSN 2318-2083 (eletrônico)

OTAVIO GOLDONI QUINA DE ALMEIDA

*Universidade do Estado do Rio de Janeiro,
UERJ, Rio de Janeiro, RJ, Brasil.*

MAIKE BARRETO

*Universidade do Estado do Rio de Janeiro,
UERJ, Rio de Janeiro, RJ, Brasil.*

SARAH SCHAYDER

*Universidade do Estado do Rio de Janeiro,
UERJ, Rio de Janeiro, RJ, Brasil.*

PEDRO MUNIZ

*Universidade do Estado do Rio de Janeiro,
UERJ, Rio de Janeiro, RJ, Brasil.*

DAVIDSON OLIVEIRA

*Universidade do Estado do Rio de Janeiro,
UERJ, Rio de Janeiro, RJ, Brasil.*

JULYA ALCANTARA

*Universidade do Estado do Rio de Janeiro,
UERJ, Rio de Janeiro, RJ, Brasil.*

MOISES RANGEL

*Universidade do Estado do Rio de Janeiro,
UERJ, Rio de Janeiro, RJ, Brasil.*

*Recebido em junho de 2024.
Aprovado em junho de 2024.*

CÁLCULOS EM RELATIVIDADE GERAL COM O USO DE PYTHON

RESUMO

A Relatividade Geral (GR) é uma das teorias mais importantes no estudo dos problemas da física moderna e é fundamental para compreender as propriedades do universo em larga escala. Como é uma teoria altamente matemática, o conhecimento em programação é frequentemente necessário para estudá-la. Embora a programação seja uma ferramenta valiosa para resolver numericamente as equações diferenciais da GR, adquirir o conhecimento necessário nas linguagens existentes exige um compromisso significativo de tempo, que poderia ser gasto diretamente na pesquisa em GR. Existem vários programas matemáticos que são mais fáceis de usar, mas a maioria deles está desatualizada ou não é gratuita. Portanto, aprender a usar um software gratuito, de fácil utilização e de código aberto torna-se extremamente importante para realizar cálculos matemáticos complexos relacionados à GR. Este é o propósito deste artigo: desenvolver uma maneira simples de usar o EinsteinPY para qualquer professor ou aluno com apenas conhecimentos introdutórios de programação. O EinsteinPY é um software desenvolvido em Python para cálculos em relatividade geral. Este software desempenha um papel crucial no ensino da Relatividade Geral tanto em nível de graduação quanto de pós-graduação.

Palavras-Chave: python; relatividade geral; einsteinpy.

CALCULATING GENERAL RELATIVITY USING PYTHON

ABSTRACT

General Relativity (GR) is one of the most important theories in the study of problems in modern physics and is fundamental to understanding the properties of the universe on a large scale. As it is a highly mathematical theory, programming knowledge is often necessary to study it. Although programming is a valuable tool for numerically solving the differential equations of GR, acquiring the necessary knowledge in existing languages requires a significant time commitment, time that could be spent directly on GR research. There are several mathematical programs that are easier to use, but most of them are either outdated or not free. Therefore, learning to use a free, user-friendly, open-source software becomes extremely important for performing complex mathematical calculations related to GR. This is the purpose of this article: to develop a simple way to use EinsteinPY for any teacher or student with only introductory programming knowledge. EinsteinPY is a software developed in Python for calculations in general relativity. This software plays a crucial role in teaching General Relativity at both undergraduate and graduate levels.

Keywords: python; general relativity; einsteinpy.

Revista UNILUS Ensino e Pesquisa

Rua Dr. Armando de Salles Oliveira, 150
Boqueirão - Santos - São Paulo
11050-071

<http://revista.lusiada.br/index.php/ruep>
revista.unilus@lusiada.br

Fone: +55 (13) 3202-4100

INTRODUÇÃO

A Relatividade Geral (RG) é uma das teorias mais importantes no estudo dos problemas da física moderna e é fundamental para entender as propriedades do universo em grande escala. Como é uma teoria muito matemática, muitas vezes é necessário conhecimento em programação para estudá-la.

Mesmo que a programação seja uma ferramenta valiosa para a resolução numérica de equações diferenciais da RG. Obter o conhecimento necessário nas linguagens existentes, exige uma dedicação de tempo grande, tempo este, que poderia ser gasto diretamente na pesquisa em RG.

Existem diversos programas matemáticos de uso mais simples, mas a maioria deles ou está desatualizado ou não são gratuitos. Então, o aprendizado sobre o uso de um software gratuito, simples de usar e de código aberto, se torna de extrema importância na realização de cálculos matemáticos complexos ligados a RG. Este é o propósito deste artigo, desenvolver uma forma simples de uso do EinsteinPY por qualquer professor ou estudante com apenas conhecimento introdutório de programação. O EinsteinPY é um software desenvolvido em Python para cálculos em relatividade geral. Este software tem uma grande importância no ensino de Relatividade Geral para graduação e pós-graduação.

Desta forma este artigo está dividido em duas partes, a primeira ensina o leitor a instalar o EinsteinPY em seu computador ou usá-lo diretamente de uma máquina virtual. A segunda parte traz um exemplo de uso do software para cálculo das principais grandezas para um espaço-tempo simples como o de Schwarzschild.

LINGUAGEM PYTHON E O EINSTEINPY

Python é uma linguagem de programação de alto nível, interpretada e de propósito geral. Criada por Guido van Rossum e lançada pela primeira vez em 1991, ela se destaca por sua sintaxe simples e legibilidade, o que a torna uma escolha popular entre iniciantes e desenvolvedores experientes.

Uma das principais características do Python é sua ampla biblioteca padrão, que oferece suporte a uma variedade de tarefas, desde manipulação de strings até desenvolvimento web e ciência de dados. Além disso, a comunidade Python é muito ativa, fornecendo uma vasta gama de bibliotecas de terceiros que estendem ainda mais a funcionalidade da linguagem. Sua flexibilidade e facilidade de aprendizado a tornam uma escolha popular tanto para projetos pequenos quanto para grandes.

Já o EinsteinPy é uma biblioteca Python especializada em relatividade geral e física matemática. Desenvolvida com o objetivo de facilitar a compreensão e exploração dos conceitos da teoria da relatividade de Einstein, a biblioteca fornece ferramentas para a realização de cálculos e visualizações relacionados a espaços-tempo curvos, buracos negros, ondas gravitacionais e outros fenômenos astrofísicos.

O EinsteinPy oferece uma variedade de funcionalidades, incluindo a criação de métricas espaço-tempo, cálculos de geodésicas para descrever o movimento de partículas em campos gravitacionais, e a simulação de sistemas astrofísicos complexos. Além disso, a biblioteca facilita a visualização desses conceitos por meio de gráficos e animações interativas.

Apesar de sua ampla funcionalidade, o EinsteinPY não fornece uma interface amigável ao usuário sem um conhecimento avançado de programação. Por este motivo, um dos intuítos deste artigo é ensinar o usuário médio a conseguir utilizar este recurso de código aberto, tanto para melhorar a didática no ensino de RG, quanto na pesquisa.

A fim de utilizarmos a biblioteca EinsteinPy, primeiramente precisamos instalar a linguagem Python em nosso ambiente de trabalho. Além disso, precisamos de um editor de texto para escrevermos os códigos, e a biblioteca. Para isso, temos três métodos distintos e conflitantes entre si, portanto, é recomendado que use apenas um dos métodos de instalação. O primeiro método consiste em instalar a linguagem pelo site

da mesma, em seguida, instalar a biblioteca com o gerenciador de pacotes 'pip' da linguagem python através do terminal do seu computador, com o VSCode como fonte para utilizar o EinsteinPY de forma simples.

A alternativa a este método é a utilização do Anaconda, que é uma plataforma de distribuição do Python, a qual facilita a criação de programas na linguagem, além disso, por possuir seu próprio gerenciador de pacotes, 'conda', o qual será usado para instalar o EinsteinPY neste método de instalação da linguagem python.

Já o terceiro método consiste em utilizar o EinsteinPY diretamente pelo Jupiter Notebook, de forma completamente online.

Após a instalação da linguagem e da biblioteca, precisamos instalar um editor de texto para escrevermos os códigos (Obs: Esse guia foi baseado para instalações no sistema operacional Windows 10).

Método de utilização via VSCode

Instalando o Python

O primeiro passo para instalar o Python é acessar o site oficial <https://www.python.org/> e entrar na seção download, e clicar em Download Python Após o download ser concluído, execute o programa de instalação.

Na primeira parte, marque a opção "Add Python 3.10 to PATH". Depois clique em "Install Now".

Figura 1: Instalação do Python.



Fonte: <https://www.python.org/>.

Após a instalação concluída, clique em "Disable path length limit". E depois em "Close". Pronto! O Python já está instalado.

Instalação da biblioteca EinsteinPy pelo terminal

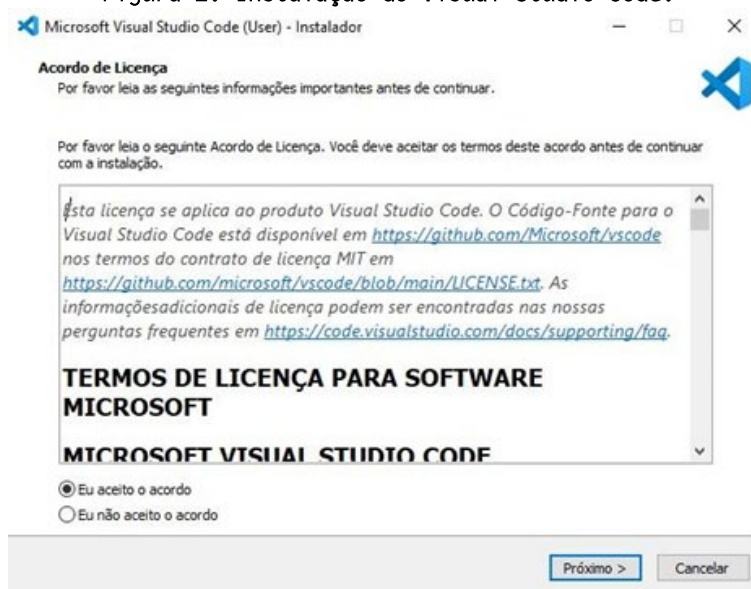
Primeiro abra o menu de pesquisa do seu desktop, digite "cmd" e abra o terminal, em seguida, digite o comando `pip install einsteinpy`. Pronto! A biblioteca EinsteinPy já está instalada.

Agora precisamos instalar o editor VScode para podermos trabalhar com os códigos e a biblioteca em si.

Instalação do Visual Studio Code (VSCode)

Primeiro acesse o site <https://code.visualstudio.com/> e clique em "Download for Windows". Após o download ser concluído, execute o programa de instalação. Na primeira parte, marque a opção "Eu aceito o acordo" e clique em "Próximo"

Figura 2: Instalação do Visual Studio Code.



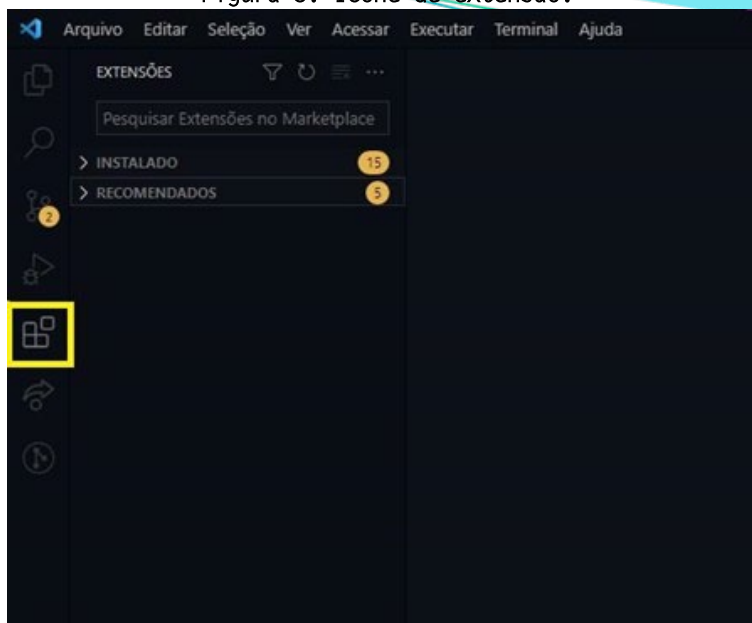
Fonte: <https://code.visualstudio.com/>.

Depois em "Próximo" novamente para confirmar o local de instalação (é recomendado que mantenha como o padrão). Depois é recomendado que deixe a pasta padrão "Visual Studio Code" selecionada e clique novamente em "Próximo". Em seguida, marque todas as opções e clique em "Próximo" mais uma vez. Clique em "Instalar" para iniciar a instalação do Visual Studio Code. Após a instalação estar completa, clique em "Concluir". Pronto! O Visual Studio Code já está instalado!

Para começarmos a usar o VSCode, é necessário também a extensão do Python no mesmo.

Primeiro, inicialize o Visual Studio Code e, então, clique no ícone da extensão localizado na barra lateral esquerda.

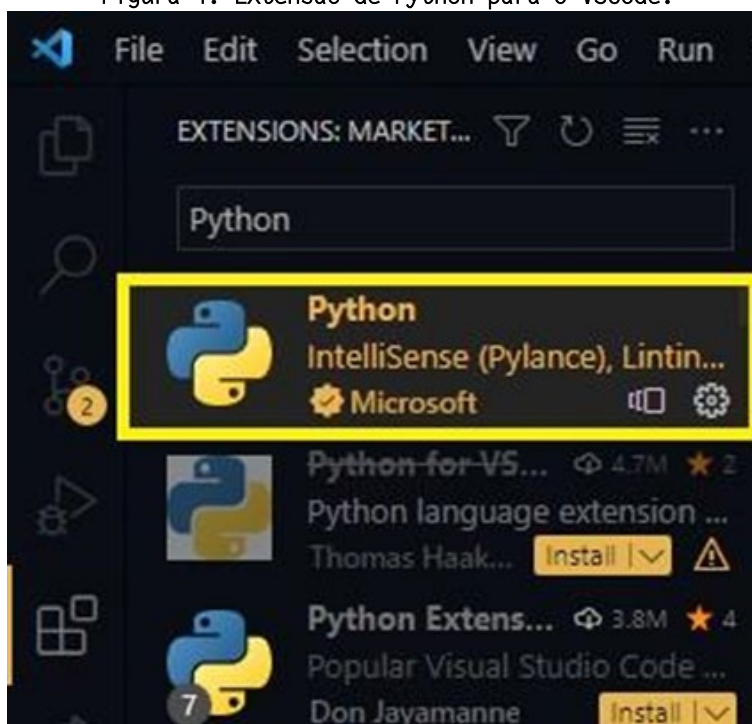
Figura 3: Ícone de extensão.



Fonte: <https://code.visualstudio.com/>.

Em seguida, clique na barra de pesquisa e digite "Python". Então, aparecerá uma opção de extensão chamada Python e com um ícone de verificado bem pequeno ao lado de Microsoft como ilustrado na figura 4. Você deve clicar nessa opção.

Figura 4: Extensão de Python para o VSCode.



Fonte: <https://code.visualstudio.com/>.

Em seguida, aparecerá uma página ao lado direito dessa barra, que contém os resultados da pesquisa, com informações dessa extensão do python e com um botão de "instalar". Clique nesse botão para que a extensão seja instalada.

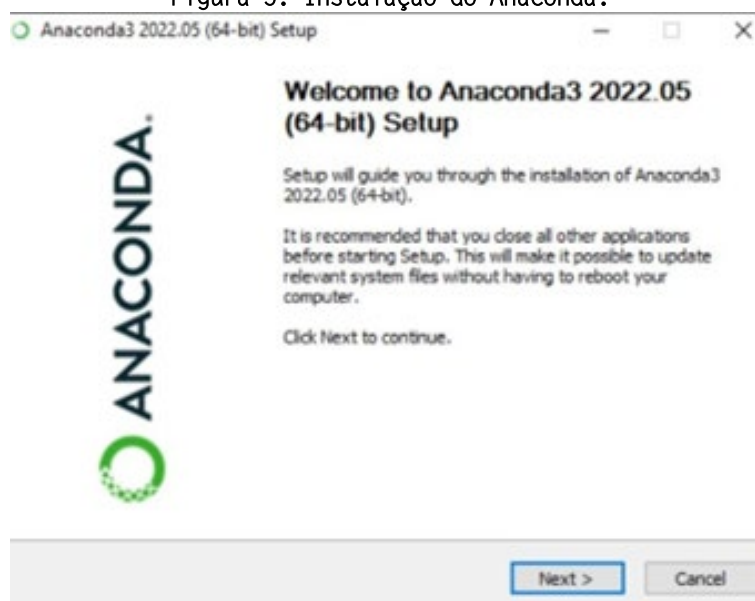
Depois de instalada, ela será habilitada imediatamente, sozinha, e seu VSCode já estará pronto para utilizar a linguagem python. Caso necessário, você pode encontrar sua nova extensão clicando no ícone da figura 3 ele se encontrará na aba de "instalados". *OBS: Antes de prosseguirmos vale explicar que, caso queira trabalhar exclusivamente com o VSCode não é necessário utilizar o segundo método, esse guia demonstra algumas alternativas para a mesma funcionalidade.*

Utilização via Anaconda

Instalação do Anaconda

Primeiro acesse o site oficial Anaconda <https://www.anaconda.com/>, entre na seção "Products -> Anaconda Distribution". E clique em download. Após o download ser concluído, execute o programa de instalação. Na primeira parte, clique em "Next".

Figura 5: Instalação do Anaconda.



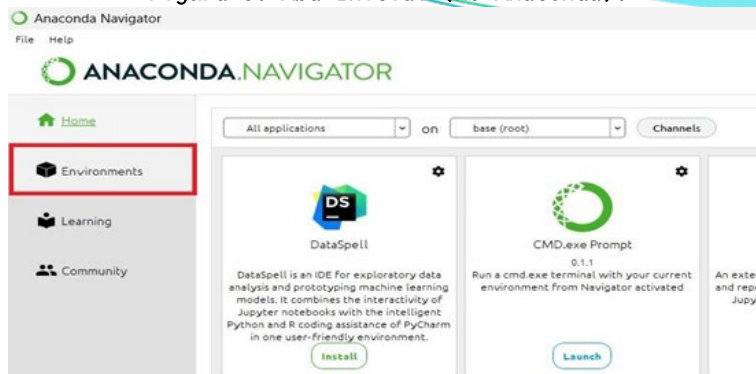
Fonte: <https://www.anaconda.com/>.

Em seguida, clique em "I agree". Mantenha a opção "Just Me(recommended)" marcada e depois clique novamente em "Next". Em seguida clique em "Next" mais uma vez para confirmar o local de instalação (é recomendado que mantenha como o padrão). É recomendado que não marque nenhuma opção nessa seção. Clique em "Install". Após a instalação estar completa, clique mais duas vezes em "Next". Em seguida, desmarque as opções selecionadas e clique em "Finish" para concluir a instalação. O Anaconda já está instalado!

Instalação da biblioteca EinsteinPy pelo cmd do Anaconda

Primeiro, abra o local onde o Anaconda foi instalado e inicie o Anaconda Navigator. Em seguida, clique em Environments.

Figura 6: Aba Inicial (no Anaconda).



Fonte: <https://www.anaconda.com/>.

Dentro da aba “Environments”, clique na seta (em “base root”) com o botão esquerdo, após “open terminal”. Digite o seguinte código “conda install -c conda-forge einsteinpy”, em seguida clique em “Enter” para prosseguir.

Em “package plan”, o comando procura o pacote e diz se o achou, logo após aparece uma pergunta se deseja proceder com a instalação da biblioteca, para proceder digite “y”.

O comando iniciará o download da biblioteca, em seguida para testar se a instalação ocorreu corretamente abre-se um terminal com python, tal como no início desta seção, clicando na seta em “base root” e após em “open with python”. Aqui basta colocar no terminal aberto com python “import einsteinpy”, caso não dê nenhum erro, a instalação foi feita perfeitamente.

OBS.: Vale ressaltar que nesse método de instalação a biblioteca funcionará em qualquer instância do Anaconda.

Utilização online via Jupyter Notebook

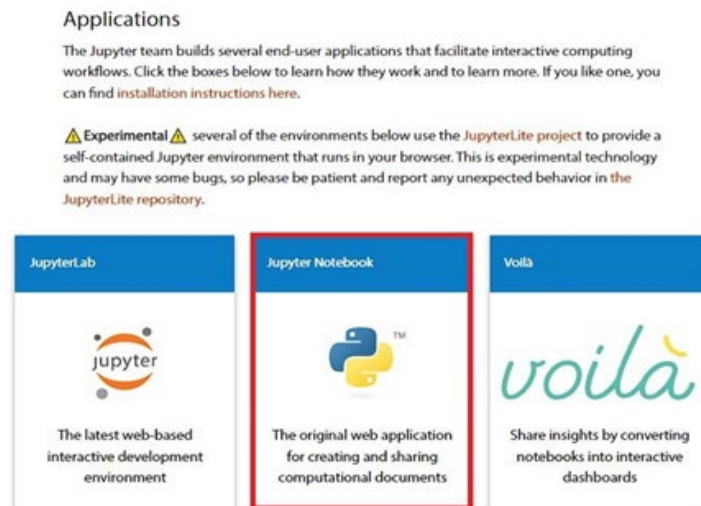
Abrindo o Jupyter Notebook

Além dos métodos anteriores, também é possível usar a biblioteca do EinsteinPY através do navegador utilizando o Jupyter Notebook, que é um aplicativo web para criar e compartilhar documentos computacionais. Ele oferece uma experiência simplificada e centrada em documentos.

Para usá-lo basta abrir o navegador em seu computador e, então, clicar no link <https://jupyter.org/>.

Em seguida, clique em “Try” no canto superior direito do site. Depois, clique em “Jupyter Notebook” na parte inferior da página.

Figura 7: Card Jupyter Notebook.



Fonte: <https://jupyter.org/>.

Em seguida, clique em “File”, depois em “New” e, finalmente, em “Notebook”.

Pronto, desta forma você consegue utilizar o EinsteinPY diretamente pelo Jupyter Notebook, completamente online. Lembrando que é necessário que a biblioteca do Python e do EinsteinPY precisa estar instalada na máquina que você está utilizando.

EXEMPLOS DA UTILIZAÇÃO DO EINSTEINPY

Uma vez que o EinsteinPY esteja corretamente instalado através de um dos métodos descritos na sessão 2. É possível utilizá-lo como uso didático, no site do projeto, <https://einsteinpy.org/>, onde o usuário encontra uma série de exemplos para a aplicação direta do programa. Cálculo das principais grandezas em métricas de espaço-tempo conhecidas como: Schwarzschild, Kerr e Kerr-Newman.

Estes exemplos são de fundamental importância nas disciplinas de RG, lecionadas na graduação e na pós-graduação em Física. Atualmente estamos esperando resultados de nossas parcerias com professores do Instituto de Física da UERJ e da UFRJ, para melhorarmos nosso material.

A título de exemplificação, colocamos o código completo para obtenção de resultados no espaço-tempo de Schwarzschild:

```
# Imports necessários
import sympy as sy
from sympy import *
from sympy import symbols, sin, Symbol, Function, var, exp, Eq, dsolve, simplify, separatevars
from sympy.solvers.ode.systems import dsolve_system
from sympy.solvers.pde import pdsolve

from einsteinpy.symbolic import (MetricTensor,
                                RicciTensor, RicciScalar,
                                RiemannCurvatureTensor)
from einsteinpy.symbolic.predefined import Schwarzschild

# Função para formatar a saída
sy.init_printing()
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = 'all'

# Definindo as variáveis da métrica
syms = sy.symbols("t r theta phi")
t, r, th, ph = syms
lamb = Function('lambda')(r)
nu = Function('nu')(r)
```



```

exp_l = exp(lamb)
exp_n = exp(nu)

# Define o tensor métrico
m = sy.diag(
    exp_l,
    -exp_n,
    -r**2,
    -(r*sin(th))**2).tolist()
metric = MetricTensor(m, syms)
metric.tensor()

# Cálculo do tensor de ricci
ricci = RicciTensor.from_metric(metric)
ricci.tensor()
ricci_tensor = ricci.tensor()
ricci_scalar = RicciScalar.from_metric(metric)
expr = metric.tensor()*1/2*ricci_scalar.expr
expr
eqs = [(Eq(ricci_tensor[i, i], 0)) for i in range(2)]
eqs[0]
eqs[1]

# Resolvendo as equações de Einstein para encontrar a forma clássica do tensor métrico de Schwarzschild
nu_ = simplify(dsolve(eqs[0],nu))
nu_
eq_l_lamb = (eqs[1].subs(nu, nu_.args[1]))
eq_l_lamb
new_lamb = dsolve(eq_l_lamb, lamb).doit()
new_lamb
Eq(exp(lamb), (exp(new_lamb.args[1])))
new_nu = simplify(nu_.subs(lamb, new_lamb.args[1]))
new_nu
Eq(exp(nu), (exp(new_nu.args[1])))
new_eqs = eqs[:2]
new_eqs
new_expr = [None, None]
new_expr[0] = simplify(expr[0,0].subs(nu, new_nu.args[1]).subs(lamb, new_lamb.args[1]))
new_expr[1] = simplify(expr[1,1].subs(nu, new_nu.args[1]).subs(lamb, new_lamb.args[1]))
new_expr
new_eqs[0] = simplify((eqs[0].subs(nu, new_nu.args[1]).subs(lamb, new_lamb.args[1])).args[0] - new_expr[0])
new_eqs[1] = simplify((eqs[1].subs(nu, new_nu.args[1]).subs(lamb, new_lamb.args[1])).args[0] - new_expr[1])
new_eqs = [simplify(Eq(eq,0)) for eq in new_eqs]
new_eqs[0]
new_eqs[1]
    
```

O código acima escreve a métrica esférica mais geral para vácuo e resolve as equações de Einstein, obtendo a métrica de Schwarzschild, que pode ser encontrada em diversos livros didáticos (D'INVERNO, 1992), dissertações, teses e artigos em revistas científicas. Exemplificando assim como o programa EinsteinPY é poderoso em aspectos didáticos para o ensino de RG.

CONCLUSÃO

Neste trabalho podemos ver que a computação pode auxiliar diretamente no ensino em outras áreas de conhecimento, como a física e a matemática. O desenvolvimento de programas que realizam cálculos complexos, muitas vezes exige um alto conhecimento de programação do usuário que muitas vezes não possui esse conhecimento. Então o uso de softwares gratuitos e simples podem agir como facilitadores. Claro, que a interface ainda não é tão simples ao usuário final, para isso é necessário o fornecimento de guias como este, para que o usuário possa concentrar suas habilidades apenas na área de conhecimento que tem interesse.

Em uma próxima etapa do trabalho desenvolveremos uma página de web para utilização do EinsteinPY sem nenhum tipo de instalação e digitação de códigos, fazendo que o usuário final possa utilizar os recursos do EinsteinPY diretamente em suas aulas e estudos.

AGRADECIMENTOS

Agradeço ao apoio financeiro da UERJ, através do projeto prodocência, sem o qual esse trabalho não seria realizado de forma tão eficiente.

REFERÊNCIAS

ANACONDA, Software aberto. Disponível em: <https://www.anaconda.com/>. Acesso em: 16 mai. 2023.

D'INVERNO, R. Introducing Einstein's Relativity. Oxford, England: Clarendon Press, 1992.

EINSTEINPY, Software aberto. Disponível em: <https://einsteinpy.org/> .Acesso em: 16 mai. 2023.

JUPYTER NOTEBOOK, Software aberto. Disponível em: <https://jupyter.org/>. Acesso em: 16 mai. 2023.

PYTHON, Software aberto. Disponível em: <https://www.python.org/>. Acesso em: 16 mai. 2023.

VISUAL STUDIO CODE, Software aberto. Disponível em: <https://code.visualstudio.com/> . Acesso em: 16 mai. 2023.